CLAIMS

What is claimed is:

1. A method, comprising:

executing a pre-boot application within an emulated pre-boot environment to test functionality of the pre-boot application, the emulated pre-boot environment executing within a user mode of a processor of a processing system during an operating system ("OS") runtime of the processing system; and

interacting with a hardware device of the processing system in response to the executing of the pre-boot application via a kernel proxy agent executing in a kernel mode of the processor.

2. The method of claim 1, further comprising reporting whether an error occurred during the interacting with the hardware device.

3. The method of claim 1, further comprising executing an interface translator in the user mode of the processor during the OS runtime to pass a request for hardware interaction from the pre-boot application to a corresponding OS user mode application programming interface ("API") if the corresponding OS user mode API is provided by the OS and to pass the request for hardware interaction to the kernel proxy agent if the OS does not provide the corresponding OS user mode API.

4. The method of claim 3 wherein interacting with the hardware device via the kernel proxy agent comprises publishing an interface to the user mode of the processor by the kernel proxy agent, the interface to enable the interface translator to pass the request for hardware interaction to the kernel proxy agent.

5. The method of claim 1, further comprising executing the kernel proxy agent to reserve a hardware resource of the processing system for use by the hardware device.

6. The method of claim 5 wherein the hardware device comprises a peripheral component interconnect ("PCI") add-in card and wherein the hardware resource includes a PCI slot for communicatively coupling the PCI add-in card into the processing system.

7. The method of claim 1 further comprising copying the pre-boot application into an option read only memory ("ROM") of the hardware device after the functionality of the pre-boot application is determined to be correct.

8. The method of claim 7 wherein the pre-boot application comprises a hardware driver of the hardware device.

9. A machine-accessible medium that provides instructions that, if executed by a machine, will cause the machine to perform operations comprising:

executing an emulated pre-boot environment to test functionality of a pre-boot application, the emulated pre-boot environment executing within a user mode of a processor of a processing system during an operating system ("OS") runtime of the processing system; and

interacting with a hardware device of the processing system in response to a request for hardware interaction from the pre-boot application, the interacting via a kernel proxy agent executing in a kernel mode of the processor.

10. The machine-accessible medium of claim 9, further providing instructions that, if executed by the machine, will cause the machine to perform further operations, comprising:

reporting whether an error occurred during the interacting with the hardware device.

11. The machine-accessible medium of claim 9, further providing instructions that, if executed by the machine, will cause the machine to perform further operations, comprising:

executing an interface translator in the user mode of the processor during the OS runtime to pass the request for hardware interaction from the pre-boot application to a corresponding OS user mode application programming interface

("API") if the corresponding OS user mode API is provided by the OS and to pass the request for hardware interaction to the kernel proxy agent if the OS does not provide the corresponding OS user mode API.

12. The machine-accessible medium of claim 11, further providing instructions that, if executed by the machine, will cause the machine to perform the operations wherein interacting with the hardware device via the kernel proxy agent comprises publishing an interface to the user mode of the processor by the kernel proxy agent, the interface to enable the interface translator to pass the request for hardware interaction to the kernel proxy agent.

13. The machine-accessible medium of claim 9, further providing instructions that, if executed by the machine, will cause the machine to perform further operations, comprising:

executing the kernel proxy agent to reserve a hardware resource of the processing system for use by the hardware device.

14. The machine-accessible medium of claim 13 wherein the hardware device comprises a peripheral component interconnect ("PCI") add-in card and wherein the hardware resource includes a PCI slot for communicatively coupling the PCI add-in card into the processing system.

15. The machine-accessible medium of claim 9, further providing instructions that, if executed by the machine, will cause the machine to perform further operations, comprising:

installing the kernel proxy agent into the kernel mode during an initialization phase of the OS.

16. A processing system, comprising:

a processor to execute an operating system ("OS") and to execute a pre-boot application, the processor having a user mode and a kernel mode;

a hardware device communicatively coupled to the processor; and

a data storage unit communicatively coupled to the processor and having stored thereon a pre-boot environment module and a kernel proxy agent, the pre-boot environment module to be executed by the processor to generate an emulated pre-boot environment within the user mode for executing the pre-boot application, the kernel proxy agent to be executed by the processor to enable interaction between the pre-boot application and the hardware device when the OS does not include an OS user mode application programming interface ("API") for interacting with the hardware device.

17. The processing system of claim 16 wherein the kernel proxy agent includes instructions to reserve a hardware resource of the processing system for use by the hardware device.

18. The processing system of claim 17 wherein the hardware device comprises a peripheral component interconnect ("PCI") add-in card and wherein the hardware resource includes a PCI slot for communicatively coupling the PCI add-in card to the processor.

19. The processing system of claim 16, further comprising the data storage unit having stored thereon an interface translator to be executed by the processor, the interface translator executable in the user mode to pass a request for hardware interaction from the pre-boot application to a corresponding OS user mode API if the corresponding OS user mode API is provided by the OS and to pass the request to the kernel proxy agent if the OS does not provide the corresponding OS user mode API.

20. The processing system of claim 19 wherein the kernel proxy agent is further to be executed by the processor to generate an error report if the quest for hardware interaction passed to the kernel proxy agent results in an error.